Computers, Materials & Continua
DOI: 10.32604/cmc.2023.031371
Article

Tech Science Press

Malicious URL Classification Using Artificial Fish Swarm Optimization and Deep Learning

Anwer Mustafa Hilal[1,2,*], Aisha Hassan Abdalla Hashim[1], Heba G. Mohamed[3], Mohamed K. Nour[4], Mashael M. Asiri[5], Ali M. Al-Sharafi[6], Mahmoud Othman[7] and Abdelwahed Motwakel[2]

[1]Department of Electrical and Computer Engineering, International Islamic University Malaysia 53100 Kuala Lumpur, Malaysia
[2]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
[3]Department of Electrical Engineering, College of Engineering, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[4]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia
[5]Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Saudi Arabia
[6]Department of Computer Science, College of Computers and Information Technology, University of Bisha, Saudi Arabia
[7]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
*Corresponding Author: Anwer Mustafa Hilal. Email: A.hilal@psau.edu.sa
Received: 15 April 2022; Accepted: 24 June 2022

Abstract: Cybersecurity-related solutions have become familiar since it ensures security and privacy against cyberattacks in this digital era. Malicious Uniform Resource Locators (URLs) can be embedded in email or Twitter and used to lure vulnerable internet users to implement malicious data in their systems. This may result in compromised security of the systems, scams, and other such cyberattacks. These attacks hijack huge quantities of the available data, incurring heavy financial loss. At the same time, Machine Learning (ML) and Deep Learning (DL) models paved the way for designing models that can detect malicious URLs accurately and classify them. With this motivation, the current article develops an Artificial Fish Swarm Algorithm (AFSA) with Deep Learning Enabled Malicious URL Detection and Classification (AFSADL-MURLC) model. The presented AFSADL-MURLC model intends to differentiate the malicious URLs from genuine URLs. To attain this, AFSADL-MURLC model initially carries out data preprocessing and makes use of glove-based word embedding technique. In addition, the created vector model is then passed onto Gated Recurrent Unit (GRU) classification to recognize the malicious URLs. Finally, AFSA is applied to the proposed model to enhance the efficiency of GRU model. The proposed AFSADL-MURLC technique was experimentally validated using benchmark dataset sourced from Kaggle repository. The simulation results confirmed the supremacy of the proposed AFSADL-MURLC model over recent approaches under distinct measures.

Keywords: Malicious URL; cybersecurity; deep learning; machine learning; metaheuristics; gated recurrent unit

1 Introduction

The advent of advanced intelligence technologies has brought a tremendous impact upon growth and development of marketplaces through multiple applications [1]. In modern era, it is absolutely essential for an institution to have online presence so that it can nurture itself into a prosperous and successful enterprise. As a result, internet and World Wide Web (WWW) have become a part of day-to-day operations in institutions and companies across the globe [2]. Unfortunately, the technical developments have also brought security problems along with it and these security issues are mainly targeted at scamming the final users. Such types of attacks contain prohibited websites that deal with forged goods and fraudulent activities are performed by means of cheating the end users through exchange of sensitive information. This information is accessed inappropriately with an intention to steal cash or identification of the users. At times, it is also done to establish the worst piece of code and malwares in user's appliances [3]. This is a common modality followed by different forms of attacks and in several circumstances, it is highly challenging to develop powerful software that can find cyber-security crimes [4]. Conventional security management technologies have evolved in the recent times due to exponential rise of security threats that occur through accelerated developments in information and communication technologies. It is important to note that only seasoned security experts can resolve this security issues and overcome the challenges faced due to cyberattacks. The increasing number of compromised URLs is the most important factor observed in such cyber-attack methodologies [5,6]. Uniform Resource Locators (URLs) indicates that the reports are universally available and it can be accessed across the globe through World Wide Web.

In general, malicious URLs can be identified with the help of Machine Learning (ML) technique through two steps as detailed herewith. At first, a suitable feature indication is obtained from the URL, and secondly, based on the feature identified, ML-related prediction methods are provided training to find out the malicious URLs [7,8]. The first step discussed above i.e., attaining the feature indication in which fruitful information regarding the URL is saved in a vector so that the ML methods can be implied to it. Several kinds of features have been assumed earlier such as content features, lexical features, popular features, and host-based features [9,10]. However, lexical features are the most widely used features as they have proved to yield superior outcomes and are comparatively simple to attain [11]. Lexical features briefly depict the lexical properties attained from URL string. These features involve statistic properties namely, URL length, total number of dots, and many more [12]. Moreover, Bag-of-Words (BoW) features are frequently utilized. BoWs represent either whether a specific string or word is displayed in the URL. Subsequently, each and every peculiar word in the training dataset is considered as a feature [13]. In second stage, such features are employed to train the prediction methods like support vector machines (SVMs). Further, such methodologies can also be reviewed as unclear blacklists [14].

In literature [15], a malicious URL recognition and detection system was proposed on the basis of Attention mechanism with Convolution Neural Network (CNN) and Long Short-Term Memory Network (Attention-Based CNN-LSTM). In relation to the weight from attention model, the local features, generated earlier, are fed as input to the LSTM network. Followed by, these features are successively pooled to compute the global feature of the URLs. At last, the URL is classified and detected by SoftMax function with global feature. In the study conducted earlier [16], an algorithm was suggested which employs AndroAnalyzer, a model that applies both deep learning systems and static

analysis. In this study, the analysis was conducted on original datasets comprising 7,622 applications. Further tests were also carried out on ML technique by comparing them with Deep Learning (DL) technique based on the feature vectors obtained.

Afzal et al. [17] presented a hybrid DL technique termed 'URLdeepDetect' to analyze the time-of-click for URLs and a classifier to detect malicious URLs. URLdeepDetect analyzes both semantic and lexical characteristics of a URL by employing different technologies that involve semantic vector methodologies and URL encryption to differentiate a URL as benign or malicious. Srinivasan et al. [18] developed DeepURLDetect (DURLD), in which raw URL is encoded with character-level embedding. In order to capture a variety of data in URL, the study employed a hidden layer in DL architecture. This is done so to extract the features from character level embedding. Afterwards, a non-linear activation function was applied to determine the possibilities of a URL being benign or malicious. Mondal et al. [19] designed a novel concept based on ML technique. The suggested model made use of different classifiers, for instance ensemble learning, to forecast the class probability of URLs. Further, it also employed a threshold to filter the decision of different classifications. In this study, the decisions were grouped to denote their respective class probabilities and signify the class labels with maximum class probability to conclude the decision upon unlabelled URL.

The current article develops an Artificial Fish Swarm Algorithm (AFSA) with Deep Learning Enabled Malicious URL Detection and Classification (AFSADL-MURLC) model. The aim of the presented AFSADL-MURLC model is to properly identify the existence of malicious URLs. To attain this, AFSADL-MURLC model initially pre-processes the data and makes use of Glove-based word embedding technique. Then, the created vector model is passed onto Gated Recurrent Unit (GRU) classification model to recognize the malicious URLs. Finally, in order to improve the efficacy of GRU model, AFSA is applied. The proposed AFSADL-MURLC technique was experimentally validated using benchmark dataset from Kaggle repository.

Rest of the paper is organized as follows. Section 2 introduces the proposed model and Section 3 offers information on experimental validation. At last, Section 4 concludes the paper.

2 The Proposed Model

In this study, a novel AFSADL-MURLC model has been proposed to properly differentiate the malicious URLs from genuine URLs. The proposed AFSADL-MURLC model primarily performs data pre-processing and makes use of Glove-based word embedding technique. Besides, the created vector model is then passed onto GRU classification model to recognize the malicious URLs. Finally, in order to improve the efficacy of GRU model, AFSA is applied. Fig. 1 depicts the block diagram of AFSADL-MURLC technique.

2.1 Pre-processing

In the first step, the tokenization of URLs is performed. Basic tokenizer can be used in this regard and it already exists in The Natural Language Toolkit (NLTK). Further, basic tokenizer is essentially maintained in python to work with programs that contain approaches compared with human language data. NLTK contains numerous libraries which are utilized on string as well as character to determine the semantic meaning behind these elements. Amongst the individuals' library, tokenizer library can be used for parsing the URL to distinct tokens. It can create an array for storing the token followed by parsing. The tokens of all the URLs are attached one by one from the array which are later distributed to vector method.

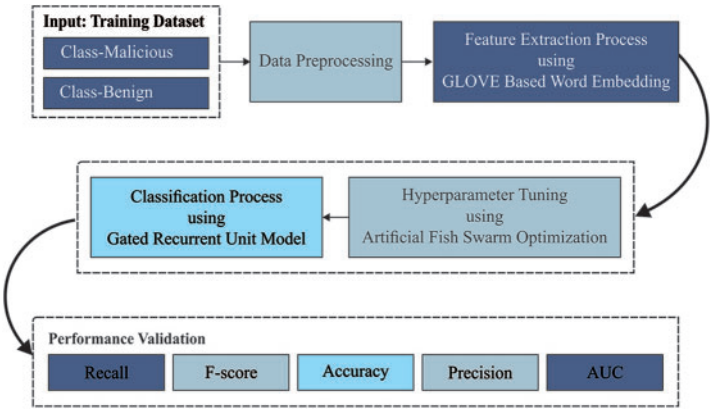

Figure 1: Overall block diagram of AFSADL-MURLC technique

2.2 Word Embedding

Global Vector (GloVe) [20], for word representation, is an unsupervised technique to derive word embedding from textual input. To start with, $A \times A$ term-based co-occurrence matrix is considered to obtain the representation. Co-occurrence matrixes are generally used in the exploration of semantic relationships amid terms. As an example, higher cosine similarity is represented between words like 'mother' and 'women' or 'queen' and 'king'. The algorithm learns from a huge Gigaword and Wikipedia corpus in an unsupervised manner. For i-th word with vector representation w_j, the objective function is represented by

$$f\left(w_j - w_j, \widetilde{w}_k\right) = \frac{P_{ik}}{P_{jk}} \tag{1}$$

In Eq. (1), P_{ik} is the probability of the instances occurring together and i and k denote the words with similar contexts. The algorithm utilizes *co*-occurrence probability as a feature by capturing the contextual word and statistics.

2.3 GRU Based Classification

At the time of classification process, the created vector model is passed onto GRU classification model to recognize the malicious URLs. GRU model holds h_t current activation, prior activation h_{t-1} and recent input x_t. Recurrent Neural Network (RNN) has the ability to learn long-term patterns and are better in comparison with feed forward Deep Neural Network (DNN) [21], since feed-forward DNN is developed in a way such that the input context features are continuously different. The hidden activation of RNN is expressed as follows using Eq. (1):

$$h_t = \varphi\left(W^h x_t + R^h h_{t-1} + b^h\right) \tag{2}$$

But a simplified RNN is hard to train using recurrent connectivity on the hidden state due to exploding or vanishing gradient problems. LSTM model addresses these difficulties by introducing cell state, input, forget and output gates to control the flow of data over a period of time. The basic principle of LSTM is that the memory cells maintain their state over a period of time. So, GRU is proposed as an alternate structure to LSTM model. GRU model is an established model in terms of efficiency than LSTM in some tasks. Following is the equation applied to the model.

$$r_t = \delta\left(W^r x_t + R^r h_{t-1} + b^r\right) \tag{3}$$

$$z_t = \delta\left(W^z x_t + R^z h_{t-1} + b^z\right) \tag{4}$$

$$\widetilde{h}_t = \varphi\left(W^h x_t + r_t \odot \left(R^h h_{t-1}\right) + b^h\right) \tag{5}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h}_t \tag{6}$$

In the above equation, hidden activation, reset and update gate values at t time are correspondingly represented by h_t, r_t and z_t. The weights used for recurrent hidden and input layer are represented by R^* and W^*, correspondingly. The bias is denoted as b^*. Tangent and sigmoid activation functions are denoted through $\varphi(\cdot)$ and $\delta(\cdot)$ respectively. There is no single memory cell in GRU compared to LSTM. Further, GRU does not have an output gate whereas it combines both forget and input gates into z_t update gate to create a balance between update activation $\widetilde{h}_t$ and prior activation h_{t-1} as demonstrated in Eq. (6). In Eqs. (5) & (6) element-wise multiplication is represented through $\odot$ term. The reset gate r_t decides whether to forget the prior activation or not (as illustrated in Eq. (5)). Fig. 2 demonstrates the structure of GRU.

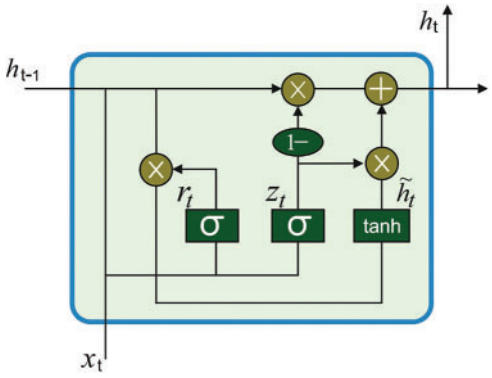

Figure 2: Architecture of GRU

2.4 AFSA Based Hyperparameter Optimization

In this final stage, AFSA is applied to improve the efficacy of GRU model by optimal-tuning of the hyperparameters [22–24]. AFSA approach is a swarm intelligence technique that is developed based on the behaviour of animals [25]. Especially, the technique has its inspiration from animal behaviours such as clustering, collision, and foraging of fish followed by collective support in a fish swarm to realize a global optimal point. Here, *Step* is determined based on the maximum distance passed in Artificial Fish (AF) technique, *Visual* is defined by the apparent distance passed through AF, *Try_The number* represents the retry amount and η represents the factor of crowd amount. Here, $X = (X_1, X_2, \ldots, X_n)$ represents the location of single AF as described by the resultant vector, and $d_{ij} = \left|X_i - X_j\right|$ represents the distance between AF i and j. Random, prey, swarm and follow are the behavioral functions of the AF.

Consider that a fish observes the food with the help of its eyes, X_i is the current location and X_j represents the arbitrarily-elected location within [0,1],

$$X_j = X_i + Visual \times rand\,(0 \sim 1) \tag{7}$$

In Eq. (8), the arbitrary value are represented by *rand*. If $Y_i > Y_j$, the fish moves in this direction. If not, a novel position X_j is arbitrarily selected to judge whether it fulfills the moving conditions as given below.

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\|X_j - X_i^t\|} \times Step \times rand\,(0{\sim}1) \tag{8}$$

Arbitrary motion can be generated using the following equation, if it does not Try_Number times.

$$X_i^{t+1} = X_i^t + Visual \times rand\,(0{\sim}1) \tag{9}$$

To avoid over-crowding, X_i i.e., an artificial present location is fixed. Then, the amount of fish in n_f company and X_c center in the region (that is. $d_{ij} < Visual$) are determined. If $Y_c/n_f < \eta \times Y_i$, the location of the companion characterizes less crowd and optimal quantity of food. The fish moves towards its companion region centre as given below.

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{\|X_c - X_i^t\|} \times Step \times rand\,(0{\sim}1) \tag{10}$$

Otherwise, it begins to implement the behaviour of prey.

In Eq. (10), X_i represents the present location of AF swarm. The swarm determines the main company Y_j as X_j in the region i.e., $d_{ij} < Visual$). If $Y_j/n_f < \eta \times Y_i$, the location of the company characterizes a lesser crowd and optimal number of food. Later, the swarm moves to X_j:

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\|X_j - X_i^t\|} \times Step \times rand\,(0{\sim}1) \tag{11}$$

It allows AF to accomplish food and company through a large regional area.

With D dimension searching space, the probable distance between two AFs is applied to limit *Visual* & *Step* of AF. *MaxD* is defined in the following equation.

$$MaxD = \sqrt{(x_{\max} - x_{\min})^2 \times D} \tag{12}$$

In Eq. (12), the lower and upper limits of the optimization range are represented by $x_{\min}$ and $x_{\max}$ correspondingly and the dimension of the search space is represented by D.

3 Experimental Validation

In this section, the proposed AFSADL-MURLC model was experimentally validated using a benchmark dataset sourced from Kaggle repository (available at https://www.kaggle.com/datasets/siddharthkumar25/malicious-and-benign-urls). In this study, the authors used 5,000 samples under benign category and 5,000 samples under malicious category.

The confusion matrices generated by the proposed AFSADL-MURLC model on identification of malicious URLs are given in Fig. 3. For 80% of training (TR) data, the proposed AFSADL-MURLC model recognized 3,924 samples under benign class and 3,936 samples under malicious class. In line with this, for 20% of testing (TS) data, AFSADL-MURLC technique recognized 990 samples under benign class and 985 samples under malicious class. Along with that, for 70% of TR data, the presented AFSADL-MURLC approach recognized 3,475 samples under benign class and 3,500 samples under malicious class. At last, on 30% of TS data, the proposed AFSADL-MURLC system recognized 1,510 samples under benign class and 1,478 samples under malicious class.

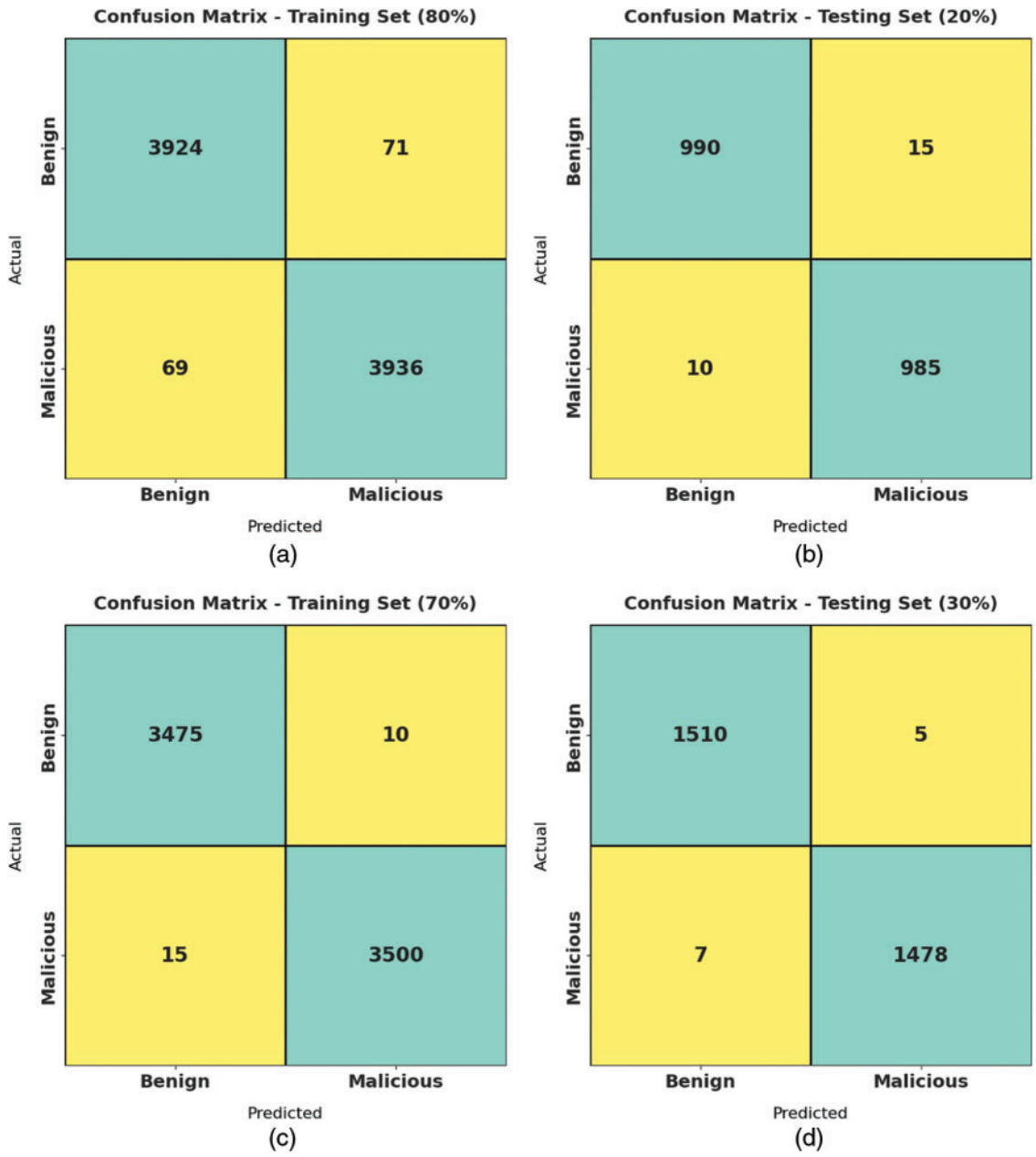

Figure 3: Confusion matrices generated by AFSADL-MURLC technique for (a) 80% of TRS, (b) 20% of TS set, (c) 70% of TRS, and (d) 30% of TS set

Tab. 1 and Fig. 4 report the overall classification outcomes accomplished by the proposed AFSADL-MURLC model with 80% of TR and 20% of TS datasets. For 80% of TR data, the presented AFSADL-MURLC model recognized the instances under benign class with $accu_y$, $prec_n$, F_{score}, and AUC values such as 98.25%, 98.27%, 98.22%, 98.25%, and 98.25% respectively. At the same time, for 80% of TR data, the proposed AFSADL-MURLC methodology recognized the instances under malicious class with $accu_y$, $prec_n$, F_{score}, and AUC values such as 98.25%, 98.23%, 98.28%, 98.25%, and

98.25% correspondingly. Moreover, on 20% of TS data, the presented AFSADL-MURLC algorithm recognized the instances under benign class with $accu_y$, $prec_n$, F_{score}, and AUC values such as 98.75%, 99%, 98.51%, 98.75%, and 98.75% correspondingly.

Table 1: Results of the analysis of AFSADL-MURLC technique under different measures on 80% of TRS and 20% of TSS

Labels	Accuracy	Precision	Recall	F-Score	AUC Score
Training set (80%)					
Benign	98.25	98.27	98.22	98.25	98.25
Malicious	98.25	98.23	98.28	98.25	98.25
Testing set (20%)					
Benign	98.75	99.00	98.51	98.75	98.75
Malicious	98.75	98.50	98.99	98.75	98.75

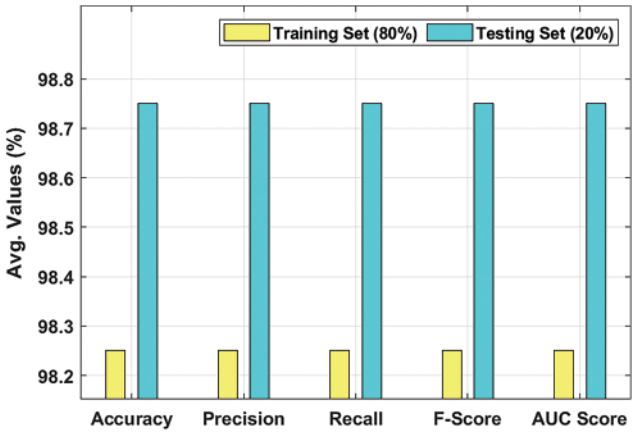

Figure 4: Results of the analysis of AFSADL-MURLC technique on 80% of TRS and 20% of TSS

Tab. 2 and Fig. 5 demonstrates the overall classification results attained by AFSADL-MURLC technique with 70% of TR and 30% of TS datasets. For 80% of TR data, AFSADL-MURLC model recognized the instances under benign class with $accu_y$, $prec_n$, F_{score}, and AUC values such as 99.64%, 99.57%, 99.71%, 99.64%, and 99.64% correspondingly. Besides, for 80% of TR data, the proposed AFSADL-MURLC approach recognized the instances under malicious class with $accu_y$, $prec_n$, F_{score}, and AUC values such as 99.64%, 99.72%, 99.57%, 99.64%, and 99.64% respectively. Furthermore, on 20% of TS data, the proposed AFSADL-MURLC technique recognized the instances under benign class with $accu_y$, $prec_n$, F_{score}, and AUC values such as 99.60%, 99.54%, 99.67%, 99.60%, and 99.60% correspondingly.

Table 2: Results of the analysis of AFSADL-MURLC technique under different measures on 70% of TRS and 30% of TSS

Labels	Accuracy	Precision	Recall	F-Score	AUC Score
Training set (70%)					
Benign	99.64	99.57	99.71	99.64	99.64
Malicious	99.64	99.72	99.57	99.64	99.64
Testing set (30%)					
Benign	99.60	99.54	99.67	99.60	99.60
Malicious	99.60	99.66	99.53	99.60	99.60

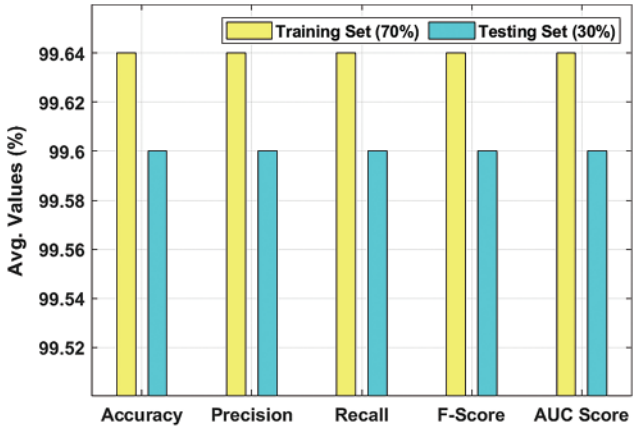

Figure 5: Results of the analysis of AFSADL-MURLC technique on 70% of TRS and 30% of TSS

Training Accuracy (TA) and Validation Accuracy (VA) values, attained by the proposed AFSADL-MURLC model on test dataset, are demonstrated in Fig. 6. The experimental outcomes imply that the proposed AFSADL-MURLC model gained the maximum TA and VA values. To be specific, VA seemed to be higher than TA.

Training Loss (TL) and Validation Loss (VL) values, achieved by the presented AFSADL-MURLC model on test dataset, are portrayed in Fig. 7. The experimental outcomes infer that AFSADL-MURLC model achieved the least TL and VL values. To be specific, VL seemed to be lower than TL.

Tab. 3 provides the results for comprehensive comparative analysis achieved by the proposed AFSADL-MURLC model and other existing models. Fig. 8 showcases the comparative investigation results attained by AFSADL-MURLC model and other existing models in terms of $accu_y$. The figure indicates that Naïve Bayes (NB) model achieved the least $accu_y$ of 95.37%. Besides, Multilayer Perceptron (MLP) and LSTM models attained slightly enhanced $accu_y$ values such as 97.94% and 98.08% respectively. In addition, Lloyd's and Random Forest (RF) models demonstrated reasonable $accu_y$ values such as 99.23% and 99.03% respectively. However, the proposed AFSADL-MURLC model accomplished superior outcomes with a maximum $accu_y$ of 99.60%.

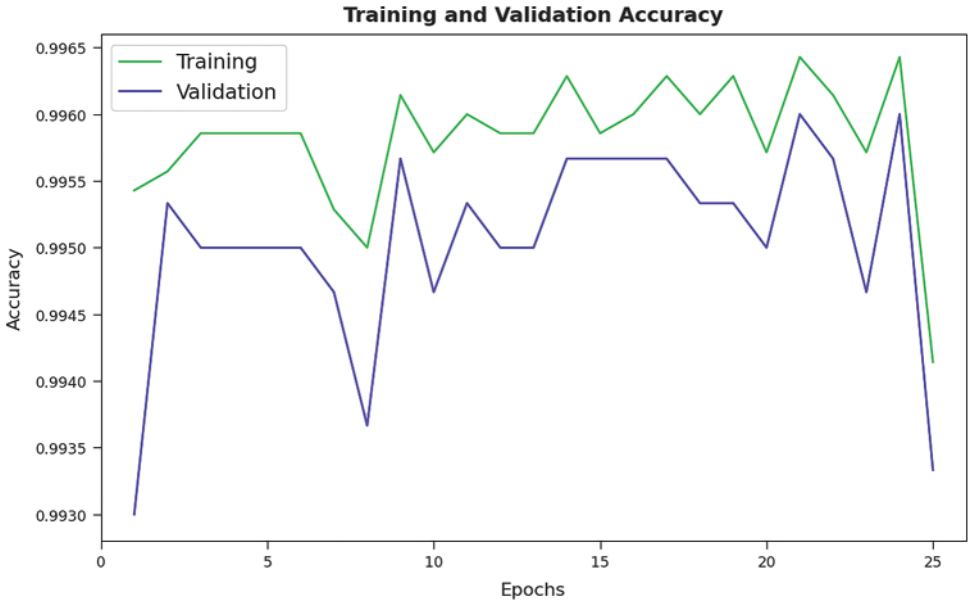

Figure 6: TA and VA analyses results of AFSADL-MURLC technique

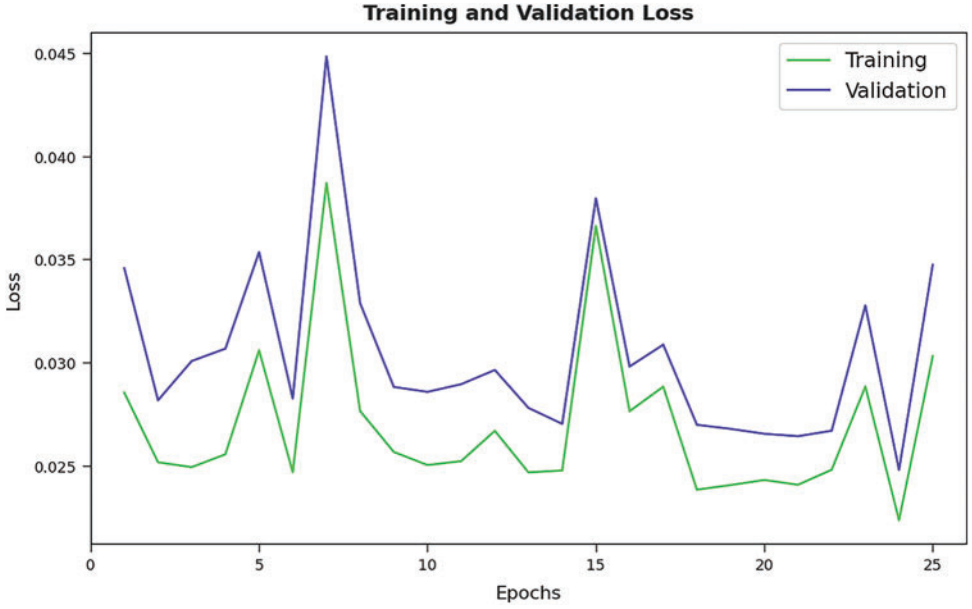

Figure 7: TL and VL analyses results of AFSADL-MURLC technique

Table 3: Comparative analysis results of AFSADL-MURLC technique and other existing approaches

Methods	Accuracy	F-Score	Precision	Recall
RF Algorithm	99.03	98.84	98.80	98.24
MLP Algorithm	97.94	98.19	99.17	97.70

(Continued)

Table 3: Continued

Methods	Accuracy	F-Score	Precision	Recall
NB Model	95.37	95.29	99.01	92.04
LSTM Model	98.08	98.02	98.90	97.51
Lloyd's Algorithm	99.23	96.70	96.90	95.51
AFSADL-MURLC	99.60	99.60	99.60	99.60

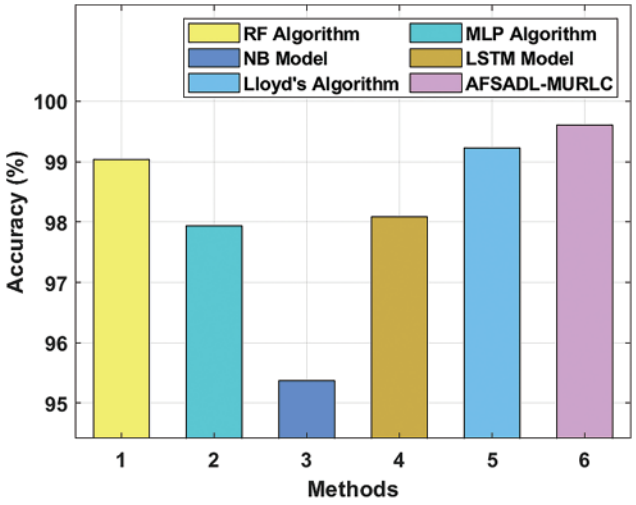

Figure 8: $Accu_y$ analysis results of AFSADL-MURLC technique and other existing approaches

Fig. 9 illustrates the comparative analysis results attained by the proposed AFSADL-MURLC technique and other existing models with respect to F_{score}. The figure infers that NB technique produced the least F_{score} of 95.29%. Also, MLP and LSTM systems achieved somewhat higher F_{score} values such as 98.19% and 98.02% respectively. Next, Lloyd's and RF models demonstrated reasonable F_{score} values such as 96.70% and 98.84% respectively. Eventually, the proposed AFSADL-MURLC technique accomplished superior outcomes with a high F_{score} of 99.60%.

Fig. 10 portrays the comparative investigation results of AFSADL-MURLC approach and other existing models in terms of $prec_n$. The figure indicates that NB algorithm produced the least $prec_n$ of 99.01%. In addition, MLP and LSTM methods achieved slightly increased $prec_n$ values such as 99.17% and 98.90% respectively. Followed by, Lloyd's and RF models demonstrated reasonable $prec_n$ values such as 96.90% and 98.80% respectively. Finally, the proposed AFSADL-MURLC methodology accomplished superior outcome with a high $prec_n$ of 99.60%.

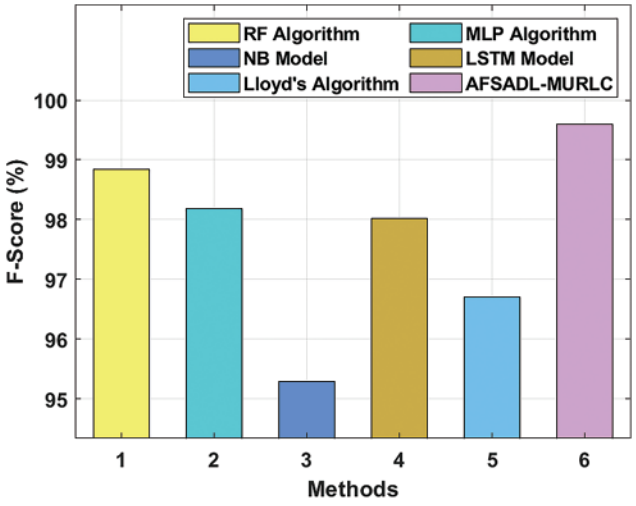

Figure 9: F_{score} analysis results of AFSADL-MURLC technique and other existing methods

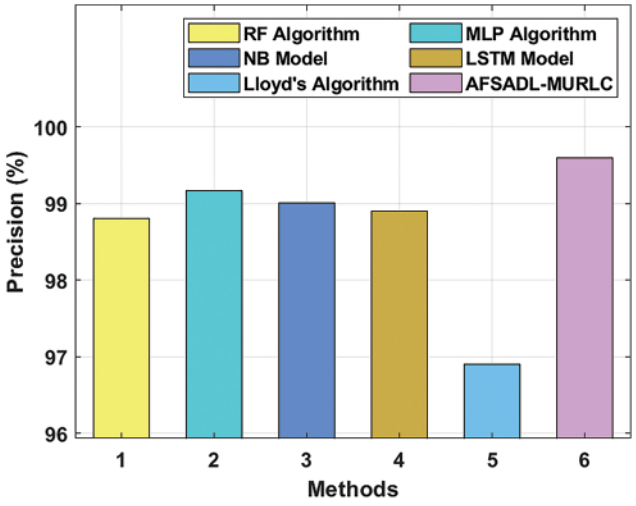

Figure 10: $Prec_n$ analysis results of AFSADL-MURLC technique and other existing methods

Fig. 11 demonstrates the comparative examination results accomplished by the proposed AFSADL-MURLC algorithm and other existing models with respect to $reca_l$. The figure implies that NB system resulted in less $reca_l$ of 98.24%. Moreover, MLP and LSTM methods achieved somewhat superior $reca_l$ values such as 97.70% and 97.51% correspondingly. In addition, Lloyd's and RF approaches demonstrated reasonable $reca_l$ values such as 95.51% and 98.24% correspondingly. At last, the proposed AFSADL-MURLC system accomplished superior outcome with a maximum $reca_l$ of 99.60%. Thus, the current study establishes that AFSADL-MURLC model has the ability to achieve maximum performance over other methods.

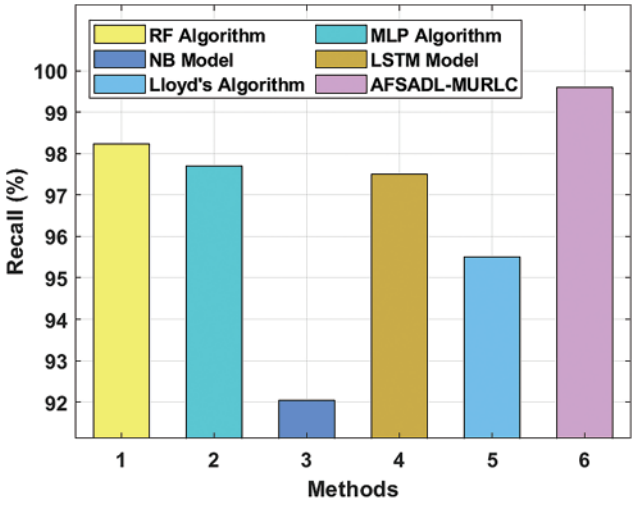

Figure 11: *Reca*$_l$ analysis results of AFSADL-MURLC technique and other existing approaches

4 Conclusion

In this study, a novel AFSADL-MURLC model has been developed to properly identify the existence of malicious URLs. The proposed AFSADL-MURLC model primarily performs data preprocessing and makes use of Glove-based word embedding technique. Besides, the created vector model is passed onto GRU classification model to recognize the malicious URLs. Finally, in order to improve the efficacy of GRU model, AFSA is applied to it. The proposed AFSADL-MURLC model was experimentally validated using benchmark dataset sourced from Kaggle repository. The simulation results confirmed the supremacy of AFSADL-MURLC model over recent approaches under distinct measures. In future, hybrid DL and metaheuristic algorithms can be designed to improve the performance in terms of malicious URL detection and classification.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (45/43). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R140), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4310373DSR21.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Johnson, B. Khadka, R. B. Basnet and T. Doleck, "Towards detecting and classifying malicious URLs using deep learning," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 11, no. 4, pp. 31–48, 2020.

[2] R. Vinayakumar, K. P. Soman and P. Poornachandran, "Evaluating deep learning approaches to characterize and classify malicious URL's," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1333–1343, 2018.

[3] Y. Liang, Q. Wang, K. Xiong, X. Zheng, Z. Yu *et al.,* "Robust detection of malicious Robust detection of malicious URLs with self-paced wide & deep learning," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021, https://doi.org/10.1109/TDSC.2021.3121388.

[4] Y. Wang, W. Cai and P. Wei, "A deep learning approach for detecting malicious JavaScript code: Using a deep learning approach to detect JavaScript-based attacks," *Security and Communication Networks*, vol. 9, no. 11, pp. 1520–1534, 2016.

[5] S. K. Birthriya and A. K. Jain, "Analysis for malicious URLs using machine learning and deep learning approaches," in *Proc. of the Int. Conf. on Paradigms of Computing, Communication and Data Sciences*, pp. 797–807, 2021, https://doi.org/10.1007/978-981-15-7533-4_63.

[6] A. A. Albraikan, S. B. Haj Hassine, S. M. Fati, F. N. Al-Wesabi, A. M. Hilal *et al.,* "Optimal deep learning-based cyberattack detection and classification technique on social networks," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 907–923, 2022.

[7] J. Yuan, Y. Liu and L. Yu, "A novel approach for malicious URL detection based on the joint model," *Security and Communication Networks*, vol. 2021, pp. 1–12, 2021, https://doi.org/10.1155/2021/4917016.

[8] A. A. Qarafi, F. Alrowais, S. Alotaibi, N. Nemri, F. N. Al-Wesabi *et al.,* "Optimal machine learning based privacy preserving blockchain assisted internet of things with smart cities environment," *Applied Sciences*, vol. 12, no. 12, pp. 1–17, 2022.

[9] Y. Liu, C. Zhu, Y. Wu, H. Xu and J. Song, "MMWD: An efficient mobile malicious webpage detection framework based on deep learning and edge cloud," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 18, pp. e6191, 2021.

[10] F. Alrowais, A. S. Almasoud, R. Marzouk, F. N. Al-Wesabi, A. M. Hilal *et al.,* "Artificial intelligence based data offloading technique for secure mec systems," *Computers, Materials & Continua*, vol. 72, no. 2, pp. 2783–2795, 2022.

[11] S. J. Bu and H. J. Kim, "Optimized URL feature selection based on genetic-algorithm-embedded deep learning for phishing website detection," *Electronics*, vol. 11, no. 7, pp. 1090, 2022.

[12] M. A. Hamza, S. B. H. Hassine, I. Abunadi, F. N. Al-Wesabi, H. Alsolai *et al.,* "Feature selection with optimal stacked sparse autoencoder for data mining," *Computers, Materials & Continua*, vol. 72, no. 2, pp. 2581–2596, 2022.

[13] J. Qiu, J. Zhang, W. Luo, L. Pan, S. Nepal *et al.,* "A survey of android malware detection with deep neural models," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–36, 2021.

[14] A. Aljofey, Q. Jiang, Q. Qu, M. Huang and J. P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from URL," *Electronics*, vol. 9, no. 9, pp. 1514, 2020.

[15] Y. Peng, S. Tian, L. Yu, Y. Lv and R. Wang, "Malicious URL recognition and detection using attention-based CNN-LSTM," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 13, no. 11, pp. 5580–5593, 2019.

[16] R. S. Arslan, "AndroAnalyzer: Android malicious software detection based on deep learning," *PeerJ Computer Science*, vol. 7, pp. e533, 2021.

[17] S. Afzal, M. Asim, A. R. Javed, M. O. Beg and T. Baker, "URLdeepDetect: A deep learning approach for detecting malicious URLs using semantic vector models," *Journal of Network and Systems Management*, vol. 29, no. 3, pp. 21, 2021.

[18] S. Srinivasan, R. Vinayakumar, A. Arunachalam, M. Alazab and K. P. Soman, "DURLD: Malicious URL detection using deep learning-based character level representations," in *Malware Analysis Using Artificial Intelligence and Deep Learning*, Springer, Cham, pp. 535–554, 2021.

[19] D. K. Mondal, B. C. Singh, H. Hu, S. Biswas, Z. Alom *et al.,* "SeizeMaliciousURL: A novel learning approach to detect malicious URLs," *Journal of Information Security and Applications*, vol. 62, pp. 102967, 2021.

[20] S. M. Mohammed, K. Jacksi and S. R. M. Zeebaree, "Glove word embedding and DBSCAN algorithms for semantic document clustering," in *2020 Int. Conf. on Advanced Science and Engineering (ICOASE)*, Duhok, Iraq, pp. 1–6, 2020.

[21] Y. Wang, W. Liao and Y. Chang, "Gated recurrent unit network-based short-term photovoltaic forecasting," *Energies*, vol. 11, no. 8, pp. 2163, 2018.
[22] D. A. Pustokhin, I. V. Pustokhina, P. Rani, V. Kansal, M. Elhoseny *et al.,* "Optimal deep learning approaches and healthcare big data analytics for mobile networks toward 5G," *Computers & Electrical Engineering*, vol. 95, pp. 1–14, 2021.
[23] G. N. Nguyen, N. H. L. Viet, M. Elhoseny, K. Shankar, B. B. Gupta *et al.,* "Secure blockchain enabled cyber-physical systems in healthcare using deep belief network with ResNet model," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 150–160, 2021.
[24] A. F. S. Devaraj, G. Murugaboopathi, M. Elhoseny, K. Shankar, K. Min *et al.,* "An efficient framework for secure image archival and retrieval system using multiple secret share creation scheme," *IEEE Access*, vol. 8, pp. 144310–144320, 2020.
[25] M. Neshat, G. Sepidnam, M. Sargolzaei and A. N. Toosi, "Artificial fish swarm algorithm: A survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 965–997, 2014.